How to troubleshoot "Error performing ASP.NET Precompilation"

This page has been made public for vendors

Question

When I run a Fortify scan using the Visual Studio plugin I get the error "Error performing ASP.NET Precompilation." How do I fix this error?

Answer

The “Error performing ASP.NET Precompilation” is commonly seen when running Fortify from Visual Studio.   Fortify performs an ASP.NET precompilation, which is not performed when a project is built within Visual Studio. Unfortunately, Fortify doesn’t provide much information to debug the problem, so you have to run Fortify sourceanalyzer from the command prompt in order to understand what is happening and what the error is. To run Fortify sourceanalyzer from the command prompt, do the following:

You’ll need to get the errors Fortify is encountering during the PreCompilation fixed, before you get any results.

1. Open the Visual Studio Command Prompt. (The Visual Studio Command Prompt loads the necessary environment variables required for the proper functioning of the Visual Studio command line tools.) You can open the VS command prompt by navigating to Start --> All Programs --> Microsoft Visual Studio <version> --> Visual Studio Tools and selecting the Developer Command Prompt.

2. Make sure you are able to build your solution successfully by running the following command (using the actual name of your solution file):

```
devenv Sample.sln /Rebuild Debug
```

If you are able to build the solution successfully, proceed to the next step. If not, fix any build errors first.

3. Next run the following Fortify SCA commands:

[Step 1: Clean]

```
sourceanalyzer -b Solution1 -clean
```

[Step 2: Translation/Build]

```
sourceanalyzer -b Solution1 -Xmx1280M -Xss8M
-debug -logfile trans.log devenv Sample.sln
/Rebuild Debug
```

[Step 3: Analysis]

```
sourceanalyzer -b Solution1 -Xmx1280M -Xss8M
-debug -logfile scan.log -scan -f Results.fpr
```

(The build ID, i.e., the argument to the "-b" option, can be anything, but it must be the same for all three commands.)

If you see the ASP.NET Precompilation error while running Step 2 for the

HPE Fortify Version	4.30 and later
Programming Language	☐ C/C++ ☑ .NET ☐ Java ☐ Objective-C ☐ Other
Fortify Audit Workbench	☐ Yes ☑ No
Fortify IDE Plugin	☑ Yes ☐ No
Other Fortify Component	☑ Yes ☐ No

Request code review tools, validations, and support HERE.

Translation/Build step, then examine the trans.log file produced by specifying the –logfile option. More details about the error should be available in the log file, to help you debug what is happening.

Additional Information

All efforts should be made to resolve the issue in the manner described above. However, if that does not work, Fortify provides a property to skip the ASP.NET Precompilation - set the com.fortify.VS.SkipASPPrecompilation to true and the com.fortify.VS.RequireASPPrecompilation property to false. However, to get an accurate scan, it is necessary to perform the precompilation step.

Fortify technical support has recommended leveraging Visual Studio itself to perform the precompilation. (Note: Visual Studio 2013 removed the ASP.NET Configuration option. If you are using 2013, please refer to this link for directions on accessing the option on that version: http://stackoverflow.com/questions/20541680/visual-studio-2013-and-asp-net-web-configuration-tool#20555928). They have suggested the following procedure:

1. Toggle debug mode for ASP compilation
    a. In the Project menu, select ASP.NET Configuration.
    b. In the Web Site Administration Tool that opens up, enter the Application tab.
    c. Click on the link, Configure Debugging and Tracing.
    d. Select Enable Debugging. (Under the covers, this adds debug="true" to the compilation element in Web.config)
2. Create a virtual directory for the Web application (if not already done)
    a. Right-click on the Web application in Solution Explorer and select Properties.
    b. In the left-hand navigation bar, click on Web.
    c. Select the Use IIS Web Server radio button.
    d. For Project URL, make up an address on localhost, e.g. http://localhost/somewebsite.
    e. Click on Create Virtual Directory.
3. Add the virtual directory as a Web site to the solution
    a. Go to File -> Add -> Existing Web Site.
    b. Select Local IIS in the left-hand navigation bar.
    c. Open the virtual directory that you created.
4. Build the Web site
    a. Right-click on the Web site and choose Build Web Site.
    b. The *.as[c|m|p]x pages should be precompiled in C:\windows\microsoft.net\framework\v2.0.50727\Temporary ASP.NET Files\somewebsite. Along with the DLLs should be PDBs as well, which are required by SCA in order to process the DLLs.
5. Point SCA to the precompiled output:
    The SCA plug-in can be made to use the precompilation output from this procedure. Simply open up <Fortify_install_dir>\Core\config\fortify.properties and uncomment the following:

    ```
    com.fortify.VS.SkipASPPrecompilation=true
    ```

    and also, set the following property to false (Default value is true)

    ```
    com.fortify.VS.RequireASPPrecompilation=false
    ```

6. Scan the application
    Re-start Visual Studio for the changes to take effect. You may now scan your ASP.NET application by clicking on the yellow Fortify button!

Note: The plug-in concatenates the following two properties to determine the full path to the precompilation output. They may be changed in fortify.properties in the event that either value is incorrect.

- com.fortify.VS.NETFrameworkRoot (default value: <.NET Framework install dir>\Temporary ASP.NET Files) The .NET Framework for VS 2010 and VS 2012 is usually installed at : C:\WINDOWS\Microsoft.NET\Framework\v4.0.30319

- com.fortify.VS.ASPVirtualRootProp (default value: name of the project)

References

- HPE Fortify Static Code Analyzer User Guide, Chapter 4: Translating .NET Source Code